

NASA TECHNICAL
MEMORANDUM

NASA TM X-3527



NASA TM X-3527 c.1

LOAN COPY: FROM
AFWL TECHNICAL LIBRARY
KIRTLAND AIR FORCE BASE



VECTORIZATION OF LINEAR
DISCRETE FILTERING ALGORITHMS

James R. Schiess

Langley Research Center

Hampton, Va. 23665





0152213

1. Report No. NASA TM X-3527		2. Government Accession No.		3. Proprietary/Security No.	
4. Title and Subtitle VECTORIZATION OF LINEAR DISCRETE FILTERING ALGORITHMS				5. Report Date July 1977	
				6. Performing Organization Code	
7. Author(s) James R. Schiess				8. Performing Organization Report No. L-11448	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665				10. Work Unit No. 505-15-37-01	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract Three existing linear filters, including the conventional Kalman filter and versions of square root filters devised by Potter and Carlson, are studied for potential application on streaming computers. The square root filters are known to maintain a positive definite covariance matrix in cases in which the Kalman filter diverges due to ill-conditioning of the matrix. Vectorization of the three filters is discussed, and comparisons are made of the number of operations and storage locations required by each filter. The Carlson filter is shown to be the most efficient of the three filters on the Control Data STAR-100 computer.					
17. Key Words (Suggested by Author(s)) Kalman filter Linear filter Vectorized methods			18. Distribution Statement Unclassified - Unlimited Subject Category 59		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 15	22. Price* \$3.50		

VECTORIZATION OF LINEAR DISCRETE FILTERING ALGORITHMS

James R. Schiess
Langley Research Center

SUMMARY

Three existing linear filters, including the conventional Kalman filter and versions of square root filters devised by Potter and Carlson (AIAA Journal, Sept. 1973), are studied for potential application on streaming computers. The square root filters are known to maintain a positive definite covariance matrix in cases in which the Kalman filter diverges due to ill-conditioning of the matrix. Vectorization of the three filters is discussed, and comparisons are made of the number of operations and storage locations required by each filter. The Carlson filter is shown to be the most efficient of the three filters on the Control Data STAR-100 computer.

INTRODUCTION

The state of the art of computers has been advanced with the recent development of streaming computers, for example, Control Data STAR-100 and Texas Instruments Advanced Scientific Computer (ASC), which process vector quantities. Because of the new capabilities of these computers, existing algorithms must be reevaluated to ensure that they perform efficiently on the new computers. The purpose of this report is to compare the relative efficiency of several sequential linear filter algorithms on the STAR-100 computer.

Since the introduction of the Kalman filter in 1960 (ref. 1), a number of optimal sequential linear filters have been proposed. The object of these filters is to estimate the state of a linear process in the presence of process and measurement noise. The Kalman filter is considered the standard method for solving these problems; various other filters are generally equivalent forms of the Kalman filter (ref. 2). However, the Kalman filter requires subtraction of nearly equal matrices in computing the covariance matrix. A not unusual result is that, after a number of measurements have been processed, the covariance matrix loses the required positive definiteness characteristics and the filter diverges.

A popular way of avoiding the degradation of the covariance matrix is the so-called square root filter. The square root filter was originally developed by James E. Potter, the equations being given in reference 3, and subsequently refined by others (ref. 4). The general approach is that the square root matrix of the covariance matrix is propagated through the measurement times; whenever the covariance matrix is desired, it is computed from the square root matrix. Computed in this way the covariance matrix is assured to be positive definite. However, the gain in computational precision for the Potter filter is compensated by a 30- to 50-percent addition in computation burden (ref. 4) on scalar computers.

More recently Carlson (ref. 5) has devised a triangular formulation of the square root filter using a Cholesky decomposition which requires approximately the same number of calculations as the Kalman filter. In this method the square root matrix is always triangular in form so that various calculations involving this matrix can be simplified.

This report presents a study of the relative efficiency of the Kalman, Potter, and Carlson filters on the Control Data STAR-100 computer at the Langley Research Center. In particular the number of operations and storage locations required are compared to determine the most efficient implementation.

SYMBOLS

$b(k+1)$	n -dimensional vector at $(k+1)$ th time, defined by equation (15)
$E\{ \}$	expectation of quantity in braces
$f(k+1)$	n -dimensional vector at $(k+1)$ th time, defined by equation (12)
$H(k+1)$	m by n measurement matrix at $(k+1)$ th time
i	denotes i th element of vector or i th column of matrix
j	denotes j th element of vector or j th row of matrix
$K(k+1)$	n by m Kalman gain matrix at $(k+1)$ th time
k	denotes quantity evaluated at k th time
L	number of results per clock (1 clock represents 40 nanoseconds)
l	length (number of components) of vector
m	number of measurements at each time
$N[\]$	condition number of matrix in brackets
n	number of state variables
$P(k)$	n by n covariance matrix at k th time
$Q(k+1)$	n by n process noise covariance matrix at $(k+1)$ th time
$R(k+1)$	m by m measurement noise covariance matrix at $(k+1)$ th time
$S(k)$	n by n square root matrix at k th time
s	start-up time, clocks
t	time for an operation, clocks

$U(k+1)$ n -dimensional process noise vector at $(k+1)$ th time
 $V(k+1)$ m -dimensional measurement noise vector at $(k+1)$ th time
 $W(k+1)$ n by n matrix at $(k+1)$ th time, defined by equation (10)
 $X(k)$ n -dimensional state vector at k th time
 $Y(k+1)$ m -dimensional measurement vector at $(k+1)$ th time
 α scalar defined by equation (13)
 γ scalar defined by equation (14)
 $\Delta Y(k+1)$ scalar measurement residual at $(k+1)$ th time
 $\phi(k+1,k)$ n by n state transition matrix from k th time to $(k+1)$ th time

Superscripts:

$+$ updated value
 T transpose of matrix
 -1 inverse of matrix

Caret (^) over a symbol indicates an estimated value.

PROBLEM STATEMENT

Many dynamical processes are mathematically modeled as discrete linear dynamical systems of equations; in the case of nonlinear processes, the models are often linearized to simplify analysis and solution of the mathematical systems. For these reasons, the state and measurement equations studied herein are given by the discrete linear system

$$X(k+1) = \phi(k+1,k) X(k) + U(k+1) \quad (k = 0, 1, 2, \dots) \quad (1)$$

where $X(k)$ is the n by 1 state vector at the k th time, $\phi(k+1,k)$ is the known n by n state transition matrix from the k th to the $(k+1)$ th time and $U(k+1)$ is the n by 1 vector of process noise at the $(k+1)$ th time. It is assumed that $U(k+1)$ is a sample from a white noise process having zero mean and known n by n covariance matrix $Q(k+1)$.

Associated with the state equations are the linear measurement equations given by

$$Y(k+1) = H(k+1) X(k+1) + V(k+1) \quad (k = 0, 1, 2, \dots) \quad (2)$$

where $Y(k+1)$ is the m by 1 measurement vector, $H(k+1)$ is the known m by n measurement matrix, and $V(k+1)$ is the m by 1 vector of measurement noise,

all three at the $(k+1)$ th time. The noise vector $V(k+1)$ is assumed to be a sample from a white noise process having zero mean and known m by m diagonal covariance matrix $R(k+1)$. It is further assumed that the process noise and measurement noise are uncorrelated and that $X(k+1)$ and $V(k+1)$ are uncorrelated.

The filtering problem is given as follows. Given a sequence of measurement vectors $Y(k)$ ($k = 1, 2, 3, \dots$), determine estimates of $X(k)$ (denoted by $\hat{X}(k)$) such that $\hat{X}(k)$ is optimal in some sense. For the filters considered herein, $\hat{X}(k)$ is the optimal estimate if the mean square error in the state is minimized.

In order to accomplish this objective, two further assumptions must be made. First, an estimate of the initial state $\hat{X}^+(0)$ is available. The statistical expected value of the state is usually used for this estimate. Second, since the covariance matrix of the state $P^+(k)$ is propagated by the filters considered, an initial value for this matrix $P^+(0)$ is also available. Generally, the initial covariance matrix can be any arbitrary positive definite matrix since the filters are not sensitive to the initial matrix used. Usually the initial matrix is chosen to be diagonal not only for simplicity but also since correlations among the state variables are unknown.

KALMAN FILTER

The Kalman filter (ref. 1) was developed to propagate the statistical mean (that is, $\hat{X}(k)$, the optimal estimate of state noted previously) and covariance of the dynamical process described by equation (1). By using the assumptions given for the process noise $U(k+1)$, the mean and covariance propagate from the k th to $(k+1)$ th time according to

$$\hat{X}(k+1) = \phi(k+1, k) \hat{X}(k) \quad (3)$$

$$P(k+1) = \phi(k+1, k) P(k) \phi(k+1, k)^T + Q(k+1) \quad (4)$$

In equation (4), $P(k+1)$ is the n by n covariance matrix at the $(k+1)$ th time and superscript T denotes matrix transpose.

Equations (3) and (4) can be used to propagate the mean and covariance throughout all time points of interest; however, neither equation makes use of information contained in the measurement vector $Y(k+1)$. The Kalman filter accomplishes this by first computing the n by m optimal gain matrix $K(k+1)$ as follows:

$$K(k+1) = P(k+1) H(k+1)^T [H(k+1) P(k+1) H(k+1)^T + R(k+1)]^{-1} \quad (5)$$

The superscript -1 indicates matrix inversion. By using the optimal gain, updated values of the state and covariance are given as

$$\hat{X}^+(k+1) = \hat{X}(k+1) + K(k+1) [Y(k+1) - H(k+1) \hat{X}(k+1)] \quad (6)$$

$$P^+(k+1) = P(k+1) - K(k+1) H(k+1) P(k+1) \quad (7)$$

The superscript + indicates values updated by incorporating the measurements at the (k+1)th time. Equations (3) and (4) are used in conjunction with equations (5) to (7) by first evaluating equations (3) and (4) with the updated state and covariance values from the previous time step. Therefore, equations (3) and (4) must initially be evaluated by using the initial estimates $\hat{X}^+(0)$ and $P^+(0)$ which were assumed available.

The matrix $P^+(k+1)$ must be a symmetric positive definite matrix since by definition of covariance $P^+(k+1) = E\{\hat{X}^+(k+1) \hat{X}^+(k+1)^T\}$ where $E\{\}$ is the statistical expectation. In theory the equations given here will propagate a symmetric positive definite matrix; however, because of the finite word length of computers, the matrix subtraction of equation (7) often yields a nonpositive unsymmetric matrix after propagation through a number of time points. The net result of this numerical problem is that the state estimate ceases to be optimal and in fact diverges from realistic values. The seriousness of this problem has prompted the development of methods to avoid the degradation of the covariance matrix.

SQUARE ROOT FILTERS

Various schemes have been developed to maintain the positive definiteness of the covariance matrix. The most numerically effective techniques are those called square root filters (ref. 5). In this approach the so-called square root matrix $S(k)$ of the covariance matrix is defined as follows:

$$S(k) S(k)^T = P(k) \quad (8)$$

Then appropriate equations are derived to propagate $S(k)$ rather than $P(k)$ through the measurement times. Whenever $P^+(k)$ is desired, it is evaluated with an equation comparable to equation (8), $P^+(k) = S^+(k) S^+(k)^T$.

The advantages of using the square root matrix are twofold. First, even in the presence of rounding errors, the product in equation (8) cannot be unsymmetric or indefinite. Second, the numerical conditioning of $S(k)$ is better than that of $P(k)$. (See ref. 4.) To show this, let $N[S(k)]$ be the condition number of $S(k)$. The condition number of $S(k)$ is defined as the ratio of the largest to the smallest eigenvalue of $S(k)^T S(k)$. Because of equation (8),

$$N[P(k)] = N[S(k) S(k)^T] = \{N[S(k)]\}^2 \quad (9)$$

On a computer with d significant binary digits, numerical difficulties can be expected when $N[P(k)]$ approaches 2^d . However, because of equation (9), the precision in computing $P(k)$ from $S(k)$ is effectively doubled since if $N[S(k)] = 2^d$ then $N[P(k)] = 2^{2d}$. In other words, with the Kalman filter, numerical difficulties may be encountered as $N[P(k)]$ approaches 2^d , but with a square root method the difficulties are encountered as $N[P(k)]$ approaches 2^{2d} .

The standard square root filter is the version derived by Potter (ref. 3) and refined by Kaminski, Bryson, and Schmidt (ref. 4). In this filter, equation (4) for propagating $P(k)$ is replaced by the following equations:

$$W(k+1) = \phi(k+1, k) S^+(k) \quad (10)$$

$$S(k+1) = [W(k+1) W(k+1)^T + Q(k+1)]^{1/2} \quad (11)$$

In equation (11), the indicated square root denotes formation of the square root matrix of the bracketed expression. The matrices $W(k+1)$, $S(k+1)$, and $S^+(k)$ are all n by n matrices. For the Potter filter, a Cholesky decomposition is applied so that the square root matrix $S(k+1)$ is a (lower or upper) triangular matrix (ref. 4). The following equations can be derived by substituting $S^+(k+1) S^+(k+1)^T$ and $S(k+1) S(k+1)^T$ for $P^+(k+1)$ and $P(k+1)$ in equations (5) to (7) and treating the case in which a scalar measurement is processed ($m = 1$):

$$f(k+1) = S^T(k+1) H_j(k+1)^T \quad (1 \leq j \leq m) \quad (12)$$

$$\alpha = R_{jj}(k+1) + f(k+1)^T f(k+1) \quad (1 \leq j \leq m) \quad (13)$$

$$\gamma = \frac{1}{\alpha + [\alpha R_{jj}(k+1)]^{1/2}} \quad (1 \leq j \leq m) \quad (14)$$

$$b(k+1) = S(k+1) f(k+1) \quad (1 \leq j \leq m) \quad (15)$$

$$S(k+1) = S(k+1) - \gamma b(k+1) f(k+1)^T \quad (1 \leq j \leq m) \quad (16)$$

$$\Delta Y(k+1) = Y_j(k+1) - H_j(k+1) \hat{X}(k+1) \quad (1 \leq j \leq m) \quad (17)$$

$$\hat{X}(k+1) = \hat{X}(k+1) + \frac{b(k+1) \Delta Y(k+1)}{\alpha} \quad (1 \leq j \leq m) \quad (18)$$

In equations (12) to (18), the m components of the measurement vector $Y(k+1)$ are processed individually as scalars. The notation defines $H_j(k+1)$ as row j of $H(k+1)$, $R_{jj}(k+1)$ as the j th diagonal element of $R(k+1)$, and $Y_j(k+1)$ as the j th component of $Y(k+1)$; the quantities α , γ , and $\Delta Y(k+1)$ are scalars. Both $f(k+1)$ and $b(k+1)$ are n -dimensional vectors.

For updating the square root matrix (refs. 4 and 5), equations (5) to (7) which update the state vector and covariance matrix are replaced by equations (12) to (18) and the following two equations:

$$\hat{X}^+(k+1) = \hat{X}(k+1) \quad (19)$$

$$S^+(k+1) = S(k+1) \quad (20)$$

One time step of the square root filter performs as follows. Equations (1), (10), and (11) are used to predict the state vector and covariance matrix. Equations (12) to (18) are cycled through once for each component of the measurement vector; during this cycling, the state and covariance are updated with information from the most recent scalar measurement. Finally equations (19) and (20) give the updated state and covariance for the entire state in preparation for the next time step.

One of the reasons the Potter filter is considerably slower than the Kalman filter is the need to cycle through equations (12) to (18) m times for each time step. During each cycle three vectors and one matrix are computed. Initially $S(k+1)$ is lower triangular, but after the first use of equation (16) this is no longer true. If it were possible to maintain the triangularity of $S(k+1)$, the number of operations required to compute the vectors and matrix could be significantly reduced. With this in mind, Carlson (ref. 5) devised a version of the square root filter which maintains the triangularity of both $S(k+1)$ and $S^+(k+1)$ at all times.

In the Carlson filter (ref. 5), the prediction equations are identical to the Potter prediction equations (eqs. (1), (10), and (11)) except that the Cholesky decomposition generates $S(k+1)$ in upper triangular form. The updating of equations (12) to (18) is replaced by the following equations:

$$f(k+1) = S(k+1)^T H_j(k+1)^T \quad (1 \leq j \leq m) \quad (21)$$

$$\alpha_0 = R_{jj}(k+1) \quad (1 \leq j \leq m) \quad (22)$$

$$b_0(k+1) = 0 \quad (1 \leq j \leq m) \quad (23)$$

$$\alpha_i = \alpha_{i-1} + f_i^2(k+1) \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (24)$$

$$a_i = \left(\frac{\alpha_{i-1}}{\alpha_i} \right)^{1/2} \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (25)$$

$$c_i = \frac{f_i(k+1)}{(\alpha_{i-1}\alpha_i)^{1/2}} \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (26)$$

$$b_i(k+1) = b_{i-1}(k+1) + S_i(k+1) f_i(k+1) \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (27)$$

$$S_i(k+1) = S_i(k+1) a_i - b_{i-1}(k+1) c_i \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (28)$$

$$\Delta Y(k+1) = Y_j(k+1) - H_j(k+1) \hat{X}(k+1) \quad (1 \leq j \leq m) \quad (29)$$

$$\hat{X}(k+1) = \hat{X}(k+1) + \frac{b_n(k+1) \Delta Y(k+1)}{\alpha_n} \quad (1 \leq j \leq m) \quad (30)$$

Equations (19) and (20) are evaluated after cycling through equations (21) to (30). In these equations, α_0 and α_i , a_i , and c_i for $(1 \leq i \leq n)$ are scalars, and $b_0(k+1)$ and $b_i(k+1)$ for $(1 \leq i \leq n)$ are n -dimensional vectors. Further, $f_i(k+1)$ is the i th component of $f(k+1)$, and $S_i(k+1)$ is the i th column of $S(k+1)$. Although the Carlson filter equations appear more complex than the Potter equations, they require fewer operations since $S(k+1)$ is always upper triangular.

Specifically, because $S(k+1)$ is upper triangular, the entries below the i th entry of the i th column are zero. Therefore, during the i th inner cycle of equations (24) to (28), the entries of $b_i(k+1)$ below the i th component need

not be computed. Further, updating the i th column of $S(k+1)$ requires computations using only the first i entries of $S_i(k+1)$ and $b_{i-1}(k+1)$. Finally, the triangular form of $S^+(k)$ and $S(k+1)$ reduces the number of operations required by equations (10) and (21). Notice that the triangular form of $S^+(k)$ does not eliminate the need for the Cholesky decomposition in equation (11) unless $Q(k+1) = 0$ (zero process noise). The reduction in computations for the Carlson filter with the triangular matrix compensates for the fact that the Carlson equations require about m times as many square roots as the Potter filter.

A comparison of the operations required by each filter for one time step on a scalar computer is given in table I, which summarizes results given in reference 5. A comparison of the three filters indicates that the Carlson filter requires fewer operations than the Potter filter. The Carlson filter is comparable to the Kalman filter for small m and n (ref. 5). An actual comparison of the timings for the filters depends on the relative times required for individual operations on a particular scalar computer.

VECTORIZATION OF FILTERS

A streaming computer such as the STAR-100 operates most efficiently by streaming strings of consecutively stored data through pipeline processing units. The increase in computational speed results from the fact that several different suboperations of a particular operation are being performed simultaneously to different data in the pipeline processor. Thus, although it takes a specified time to perform the entire operation on one number, the simultaneous assembly line processing of many numbers reduces the total required time by a considerable amount. Therefore, it is necessary to vectorize any algorithm as much as practical in order to develop an efficient procedure. An indication of this fact is given by the timing formula

$$t = s + \frac{\ell}{L} \quad (31)$$

where

t	time for operation, clocks
s	start-up time, clocks
ℓ	length of vector
L	number of results per clock

For the STAR-100 computer, 1 clock is 40 nanoseconds. From the formula (eq. (31)) it can be seen that, because of the start-up time, one long vector operation is more efficient than several short operations.

The hardware of the STAR-100 computer provides not only the basic vector operations of add and dot product but also specially defined operations such as

multiply, divide, and square root. For these latter operations, the entries of the resultant vector are the product, quotient, or square root of the corresponding entries of the operand vectors. Such operations are especially useful for algorithms which utilize matrix equations.

Application of vector operations to most matrix equations is straightforward. For example, the product of a matrix and vector, as in equation (3), yields a vector which is a linear combination of the column vectors of the matrix. For equation (3), the entries of the vector $\hat{X}(k)$ provide the coefficients of the linear combination. In a similar fashion, the product of two matrices can be written as a linear combination of column vectors from the pre-multiplying matrix with coefficients from the postmultiplying matrix. As a standard approach it is therefore assumed here that matrices are stored as column vectors.

Storing matrices by columns also produces a relatively efficient product of a matrix and transposed matrix. For example, the product $P(k+1) H(k+1)^T$ in equation (5) is defined by

$$M_{ij} = \sum_{h=1}^n P_{ih}(k+1) H_{jh}(k+1) \quad (1 \leq i \leq n; 1 \leq j \leq m) \quad (32)$$

In terms of column vectors, equation (32) can be written as

$$M_j = \sum_{h=1}^n P_h(k+1) H_{jh}(k+1) \quad (1 \leq j \leq m)$$

Computing the entire jth column is inefficient since as the sum ranges over h , $H(k+1)$ is accessed elementwise across columns. Instead it is more efficient to vary j and add to each column of M before increasing h . In this way contiguous locations of $H(k+1)$ are accessed, and physically transposing the matrix is avoided.

Since the Kalman filter consists entirely of vector and matrix equations, vectorization of this filter is straightforward. In the case of the Potter filter, vectorization is immediate although incomplete because of the scalar operations in equations (13) and (14). Since these scalar equations are unavoidable, the accompanying inefficiency must be accepted.

Efficient vectorization of the Carlson filter requires modification of two equations. The recursive generation of α_i ($0 \leq i \leq n$) requires that equations (22) and (24) remain scalar equations; however, equations (25) and (26) are easy to vectorize. Once the values of α_i are generated in a vector of length $n + 1$; the vector square root can be calculated. Then equations (24) and (25) are vector equations involving n adjacent entries of the $n + 1$ dimensional vector $(\alpha_0^{1/2}, \alpha_1^{1/2}, \dots, \alpha_n^{1/2})$. After computing the vectors $(a_1, a_2, \dots, a_n)^T$ and $(c_1, c_2, \dots, c_n)^T$, equations (27) and (28) are cycled through as before. The remainder of the Carlson equations remain unchanged.

In both the Potter and Carlson algorithms, the measurement matrix is always accessed by rows. Therefore, it is most efficient to store this matrix as row vectors for both algorithms.

The number of operations required to process one measurement vector by each of the vectorized algorithms as a function of the dimensions n and m is given in table II. The operation count does not include the operations required to evaluate the matrices $\phi(k+1,k)$, $Q(k+1)$, $H(k+1)$, and $R(k+1)$ since these matrices are problem dependent. The asterisk (*) denotes operations on vectors of length n or less; these are the operations in which the triangular matrices $S^+(k)$ and $S(k+1)$ occur. These operations then apply to vectors of average length $n/2$; this is approximately equivalent to half as many operations applied to a vector of length n (true equivalence does not exist because of the number of start-up times). On this basis the Carlson filter is faster than the Potter filter and apparently comparable to the Kalman filter.

It is of interest now to look at two specific cases of the dimensions of the filtering problem. In most realistic problems n is larger or possibly equal to m ; in other words, there are usually more unknown variables than measurements at each time point. Of particular interest are the extreme cases of scalar measurements ($m = 1$) and of large measurement vectors ($m = n$). For scalar measurements, operations on vectors of length m reduce to scalar operations. The operation counts in table III indicate that the Carlson filter may be faster than the Kalman filter because of the fewer number of scalar operations. Table IV for $m = n$ shows the Carlson filter comparable to the Kalman filter because of the fewer vector operations required by the Carlson filter. Both tables show that the Potter filter is apparently the slowest.

The number of storage locations used by the three filters are indicated in table V. These results are based on the assumptions that $R(k+1)$ is diagonal and can, therefore, be stored as an n -dimensional vector and that the matrices $Q(k+1)$, $\phi(k+1,k)$, and $P^+(k)$ can be used as temporary storage when not serving their nominal purposes. The expressions in table V do not consider savings due to sparse matrices or other characteristics which are problem dependent. The square root filters differ from the Kalman filter in that they do not require the gain matrix $K(k+1)$. The Carlson filter requires fewer storage locations than the Potter filter since $S(k+1)$ is triangular; $S(k+1)$ can be stored as a vector with entries defined by

$$S_g(k+1) = S_{ij}(k+1)$$

for

$$g = i + \frac{j(j-1)}{2}$$

Both square root filters use the same locations to store $S(k+1)$ and $S^+(k)$; both filters use $P^+(k)$ to store the quantities defined by equations (12) to (15), (17), (21) to (27), and (29). The expressions in table V do not include the locations required for counters, subscripts, and so forth.

RESULTS OF STAR VECTORIZATION

In order to investigate the three filters under specific circumstances, the filters were implemented on the Control Data STAR-100 computer. The concepts of the preceding section were applied in order to enhance the efficiency of the implementations.

Table VI presents the execution times required by the three filters to process data over 1000 time steps. The given times exclude the time required to evaluate the problem-dependent matrices $\phi(k+1,k)$, $H(k+1)$, $Q(k+1)$, and $R(k+1)$; therefore, the results can be duplicated with any identically dimensioned matrices. Aside from requiring $R(k+1)$ to be a diagonal matrix, which is representable as a vector, no particular structures are assumed for these matrices. Thus for specific problems the matrices may be sufficiently sparse to permit use of the sparse vector operations on the STAR; therefore, the speed of all the filters could be enhanced.

The values of n and m chosen for study are representative of the dimensions of most filtering problems and are comparable to the values studied by Carlson (ref. 5). With one exception, the values fulfill the common circumstance that $m \leq n$. In all but one of the dimensional problems of table VI, the Carlson filter is faster than both the Kalman filter and the Potter filter. For the small dimensional problem, the Kalman and Carlson filters are comparable. In all the other problems, including the one when $m > n$ and the ones in which scalar measurements are used, the Carlson filter is faster than the Kalman filter. The largest timing differences between the filters occur when $m = n$; these differences increase as m and n increase. Similarly, the timing differences increase with increasing n for the scalar measurement ($m = 1$) problems. Generally, therefore, on the STAR-100 computer, the Carlson filter is consistently faster than the Kalman filter except for small dimensional problems. On the other hand, in most problems the Potter filter is the slowest of the three filters; this result follows from the square structure of $S^+(k+1)$ and the extensive use of scalar arithmetic. The times for $n = 4$ and $m = 8$ suggest that the Potter filter may be more efficient when $m > n$.

CONCLUDING REMARKS

Three existing linear filters, including the conventional Kalman filter and versions of square root filters devised by Potter and Carlson (AIAA Journal, Sept. 1973), were studied for potential application on streaming computers. Of these three discrete linear filters, the two square root filters offer the advantage of greater precision since they are less susceptible to divergence due to ill-conditioning of the covariance matrix. This study has indicated that a vectorized version of the Carlson square root filter would require no more calculations than the Kalman filter and sometimes even fewer calculations. Depending on the timings of various operations on a particular computer, the Carlson filter probably requires the least time of the three filters. On the Control Data STAR-100 computer the Carlson filter was consistently the fastest method except for low dimension problems. In addition, the vectorized Carlson filter requires less computer storage than either of the other methods.

Because the Carlson filter possesses greater precision than the Kalman filter and requires fewer computer resources than the Kalman or Potter filter on a vector computer, the Carlson filter is the recommended filter.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
May 26, 1977

REFERENCES

1. Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems. Trans. ASME, Ser. D: J. Basic Eng., vol. 82, no. 1, Mar. 1960, pp. 35-45.
2. Jazwinski, Andrew H.: Stochastic Processes and Filtering Theory. Academic Press, Inc., 1970.
3. Battin, Richard H.: Astronautical Guidance. McGraw-Hill Book Co., c.1964, pp. 338-339.
4. Kaminski, Paul G.; Bryson, Arthur E., Jr.; and Schmidt, Stanley F.: Discrete Square Root Filtering: A Survey of Current Techniques. IEEE Trans. Autom. Control, vol. AC-16, no. 6, Dec. 1971, pp. 727-736.
5. Carlson, Neal A.: Fast Triangular Formulation of the Square Root Filter. AIAA J., vol. 11, no. 9, Sept. 1973, pp. 1259-1265.

TABLE I.- NUMBER OF OPERATIONS REQUIRED FOR ONE TIME STEP ON SCALAR COMPUTER

Filter	Number of operations			
	Add	Multiply	Divide	Square root
Kalman	$\frac{3}{2}n^3 + \frac{3}{2}mn^2 + \frac{1}{2}(3m-1)n$	$\frac{3}{2}n^3 + \frac{3}{2}(m+1)n^2 + \frac{7}{2}mn$	m	0
Potter	$\frac{5}{3}n^3 + 3mn^2 - \frac{2}{3}n + m$	$\frac{5}{3}n^3 + (3m+2)n^2 + \left(3m - \frac{2}{3}\right)n + m$	2m + n	m + n
Carlson	$\frac{7}{6}n^3 + \frac{1}{2}(3m+1)n^2 + \left(\frac{m}{2} - \frac{2}{3}\right)n$	$\frac{7}{6}n^3 + \left(2m + \frac{5}{2}\right)n^2 + \left(4m - \frac{2}{3}\right)n$	(m + 1)n	(m + 1)n

TABLE II.- NUMBER OF OPERATIONS REQUIRED FOR ONE TIME STEP ON VECTOR COMPUTER

Filter	Number of operations on vectors of length n				Number of operations on vectors of length m		Number of scalar operations		
	Add (a)	Multiply (a)	Divide	Square root	Add	Multiply	Add	Divide	Square root
Kalman	$2n^2 + 2(m+2)n + m^2 + 3m + 2$	$2n^2 + 2(m+2)n + m^2 + m$	0	0	$n^2 + (m+2)n + m + 2$	$n^2 + (m+1)n$	0	0	0
Potter	$3n^2 + (2m+5)n + 2m + 2$	$3n^2 + (4m+1)n + 3m$	0	0	0	0	$mn^2 + mn + 3$	2m	m
Carlson	$n^2 + 2n^*2 + 3mn^* + 3n + 2m + 2$	$n^2 + 3n^*2 + n + 3mn^* + 4m$	2m	m	0	0	$\frac{mn^2}{2} + \frac{5}{2}mn + 2m$	m	0

Asterisk () denotes operations on vectors of length between 1 and n.

TABLE III.- NUMBER OF OPERATIONS REQUIRED FOR ONE TIME STEP ON VECTOR COMPUTER WHEN $m = 1$

Filter	Number of operations on vectors of length n				Number of scalar operations		
	Add (a)	Multiply (a)	Divide	Square root	Add	Divide	Square root
Kalman	$2n^2 + 6n + 6$	$2n^2 + 6n + 2$	0	0	$n^2 + 3n + 3$	$n^2 + 2n$	0
Potter	$3n^2 + 7n + 4$	$3n^2 + 5n + 3$	0	0	$n^2 + n + 3$	2	1
Carlson	$n^2 + 2n^{*2} + 3n + 3n^{*} + 4$	$n^2 + 3n^{*2} + n + 3n^{*} + 4$	2	1	$\frac{n^2}{2} + \frac{5n}{2} + 2$	1	0

^aAsterisk (*) denotes operations on vectors of length between 1 and n .

TABLE IV.- NUMBER OF OPERATIONS REQUIRED FOR ONE TIME STEP ON VECTOR COMPUTER WHEN $m = n$

Filter	Number of operations on vectors of length n				Number of scalar operations		
	Add (a)	Multiply (a)	Divide	Square root	Add	Divide	Square root
Kalman	$7n^2 + 10n + 4$	$7n^2 + 6n$	0	0	0	0	0
Potter	$5n^2 + 7n + 2$	$7n^2 + 4n$	0	0	$n^3 + n^2 + 3$	$2n$	n
Carlson	$n^2 + 5n^{*2} + 5n + 2$	$n^2 + 6n^{*2} + 5n$	$2n$	n	$\frac{n^3}{2} + \frac{5n^2}{2} + 2n$	n	0

^aAsterisk (*) denotes operations on vectors of length between 1 and n .

TABLE V.- STORAGE LOCATIONS REQUIRED BY FILTERS

Filter	Storage locations
Kalman	$4n^2 + 2n + 2mn + 2m$
Potter	$4n^2 + 2n + mn + 2m$
Carlson	$\frac{7n^2}{2} + \frac{5n}{2} + mn + 2m$

TABLE VI.- EXECUTION TIMES ON CONTROL DATA STAR-100 FOR
VECTORIZED FILTERS TO PROCESS 1000 TIME STEPS

Dimension		Execution time, sec, for -		
n	m	Kalman filter	Potter filter	Carlson filter
32	32	161.7	134.8	128.5
32	16	91.9	98.5	87.7
32	8	68.3	80.4	67.4
32	4	59.5	71.3	57.2
32	1	53.7	64.8	49.5
16	16	38.1	31.8	31.6
16	8	21.9	23.3	21.5
16	1	13.3	15.9	12.5
8	8	9.2	7.7	8.0
8	1	3.3	4.0	3.2
4	8	5.3	3.0	3.6
4	1	.7	1.9	.7



503 001 C1 U G 770708 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546